

LAPORAN PRAKTIKUM

IMPLEMENTASI TABLE-DRIVEN DAN STATE-BASED CONSTRUCTION DALAM PEMROGRAMAN

BAB I: PENDAHULUAN

1.1 Latar Belakang Dalam pengembangan perangkat lunak, terdapat berbagai teknik pemrograman yang digunakan untuk meningkatkan efisiensi dan keterbacaan kode. Dua teknik yang sering digunakan adalah **table-driven** dan **state-based construction**. Teknik table-driven memanfaatkan tabel untuk menyimpan data yang dapat diakses secara dinamis, sementara state-based construction digunakan untuk mengelola perubahan status suatu objek dalam program.

1.2 Tujuan Praktikum ini bertujuan untuk:

1. Mempelajari dan mengimplementasikan teknik table-driven dalam pemrograman.
2. Menerapkan state-based construction dalam sebuah simulasi sederhana.
3. Memahami penggunaan Git dalam pengelolaan versi kode program.

BAB II: LANDASAN TEORI

2.1 Table-Driven Programming Teknik table-driven adalah metode pemrograman di mana tabel digunakan untuk menyimpan data yang nantinya akan diakses oleh program tanpa menggunakan banyak pernyataan bersyarat (if-else atau switch-case). Keunggulan metode ini adalah kemudahan dalam pembaruan data dan keterbacaan kode yang lebih baik.

2.2 State-Based Construction State-based construction adalah teknik pemrograman di mana suatu objek memiliki beberapa keadaan (state) dan dapat berpindah dari satu keadaan ke keadaan lain berdasarkan aksi tertentu. Teknik ini sering digunakan dalam sistem berbasis mesin keadaan (finite state machine) seperti simulasi pintu otomatis atau sistem antrian.

2.3 Git dan Version Control Git adalah sistem version control yang digunakan untuk melacak perubahan kode program. Dalam praktikum ini, Git digunakan untuk melakukan inisialisasi repository, melakukan commit, dan mengunggah kode ke repository GitHub.

BAB III: METODE PRAKTIKUM

3.1 Alat dan Bahan

1. Laptop dengan sistem operasi Windows/Linux/macOS.
2. Node.js terinstal untuk menjalankan kode program.
3. Editor kode (Visual Studio Code, Sublime Text, atau sejenisnya).
4. Git terinstal untuk version control.
5. Koneksi internet untuk mengunggah kode ke GitHub.

3.2 Langkah-Langkah Praktikum

1. Membuat proyek baru dengan format console application menggunakan Node.js.
2. Menamai proyek dengan format tpmodul3_NIM.
3. Melakukan inisialisasi repository Git dengan `git init`.
4. Mengimplementasikan teknik table-driven.
5. Mengimplementasikan teknik state-based construction.
6. Menguji kode program untuk memastikan hasil yang diharapkan.
7. Mengunggah kode ke repository GitHub.

BAB IV: IMPLEMENTASI

4.1 Implementasi Table-Driven Kode program:

```
class KodePos {
  constructor() {
    this.data = {
      "Batununggal": 40266,
      "Kujangsari": 40287,
      "Menger": 40267,
      "Wates": 40256,
      "Cijaura": 40287,
      "Jatisari": 40286,
      "Margasari": 40286,
      "Sekarjati": 40286,
      "Kebomawaru": 40272,
      "Maleer": 40274,
      "Samoja": 40273
    };
  }

  getKodePos(kelurahan) {
    return this.data[kelurahan] || "Kode pos tidak ditemukan";
  }
}
```

Output:

```
[Running] node "/Users/G I L A N G/Documents/KAMPUS/AKADEMIK/SEMESTER 4/KONSTRUKSI PERANGKAT LUNAK/FILE/PROGRAM/PERTEMUAN 4/TP/Modul 04_2311104018.js"
Kode pos Batununggal: 40266
Kode pos Kujangsari: 40287
Kode pos tidak terdaftar: Kode pos tidak ditemukan
[Done] exited with code=0 in 0.222 seconds
```

4.2

Implementasi State-Based Construction Kode program:

```
class DoorMachine {
  constructor() {
    this.state = "Terkunci";
  }

  bukaPintu() {
    if (this.state === "Terkunci") {
      console.log("Pintu masih terkunci! Tidak bisa dibuka.");
    } else {
      console.log("Pintu terbuka.");
    }
  }

  kunciPintu() {
    this.state = "Terkunci";
    console.log("Pintu terkunci.");
  }

  bukaKunci() {
    this.state = "Terbuka";
    console.log("Pintu tidak terkunci.");
  }
}
```

Output:

```
[Running] node "/Users/G I L A N G/Documents/KAMPUS/AKADEMIK/SEMESTER 4/KONSTRUKSI PERANGKAT LUNAK/FILE/PROGRAM/PERTEMUAN 4/TP/Modul 04_2311104018.js"
Kode pos Batununggal: 40266
Kode pos Kujangsari: 40287
Kode pos tidak terdaftar: Kode pos tidak ditemukan
[Done] exited with code=0 in 0.222 seconds
```

BAB V: HASIL DAN ANALISIS

5.1 Hasil Implementasi Program berhasil dijalankan dengan hasil:

- Saat memanggil `getKodePos("Batununggal")`, program mengembalikan kode pos 40266.
- Saat mengeksekusi `bukaPintu()`, program mencetak "Pintu tidak terkunci".
- Saat mengeksekusi `kunciPintu()`, program mencetak "Pintu terkunci".

5.2 Analisis

- Teknik table-driven mempermudah pengelolaan data tanpa menggunakan banyak kondisi.
- Teknik state-based construction memungkinkan implementasi finite state machine dengan lebih terstruktur.
- Penggunaan Git mempermudah pengelolaan dan kolaborasi kode program.

BAB VI: KESIMPULAN DAN SARAN

6.1 Kesimpulan Dari praktikum ini, dapat disimpulkan bahwa:

1. Teknik table-driven efektif dalam mengelola data secara dinamis.
2. Teknik state-based construction membantu dalam implementasi sistem berbasis state.
3. Penggunaan Git sangat membantu dalam version control.

DAFTAR PUSTAKA

1. Sommerville, Ian. Software Engineering (10th ed.). Pearson, 2015.
2. Gamma, E., Helm, R., Johnson, R., & Vlissides, J. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1994.
3. Dokumentasi Git: <https://git-scm.com/doc>
4. Dokumentasi Node.js: <https://nodejs.org/en/docs/>

Disusun oleh: Tegar Kang Ageng Gilang Telkom University Purwokerto – Semester 4